Accessing an Interactive GPU on the BGU Cluster

Step 1: Connect to the BGU Network (VPN)

If you are **not connected to the BGU Wi-Fi**, you must first connect via **BGU VPN**.

If you are unfamiliar with VPN:

- Search on Google: **“BGU VPN”**
- Follow the official university instructions

⚠️ Without VPN, access to the cluster from outside campus will fail.

Step 2: Connect to the Cluster Login (Master) Node

After connecting to the VPN, open a terminal and connect to the cluster login node:

```
ssh user_name@slurm.bgu.ac.il
```

Replace `user_name` with your BGU username, “enter”, then use your BGU password.

This login node is used **only to request computing resources**, do not run code here.

Step 3: Request an Interactive GPU Session

From the login node, request a GPU:

- `sinteractive --qos course --part gtx1080 --gpu 1 --time 0-08:00:00`

This command:

- Requests **1 GPU**
- Uses the **course queue**
- Allocates a **GTX 1080 Ti GPU**
- Reserves it for **up to 8 hours** (example)

If GPUs are currently busy, the command will wait until one becomes available.

Step 4: Identify and Access Your Assigned GPU Node

Once the interactive job starts, you will be assigned a **specific GPU compute node (it will print something to the terminal like the node name)**, for example:

```
user_name@ise-pheno-07.auth.ad.bgu.ac.il
```

This means your GPU is located on the node `ise-pheno-07`.

You can access this node in **two equivalent ways**:

Option A: Direct SSH Access (like we did for the slurm.bgu.ac.il)

From another terminal (or reconnecting later):

```
ssh user_name@ise-pheno-07.auth.ad.bgu.ac.il
```

This gives you direct access to the same GPU node where your job is running.

Option B: Access via an IDE (Recommended)

For convenience, you may connect to the GPU node using an **SSH-based IDE**, which provides:

- File browser (folders, files)
- Editor with syntax highlighting
- Integrated terminal
- Easier code navigation

Supported tools:

- **Visual Studio Code (Remote-SSH) - recommended for editing code, file browsing, running code etc.**
- **PyCharm (Remote Interpreter via SSH)**
- **MobaXterm - recommended for file browsing.**

Read also the next pages.

In the IDE:

Configure an SSH connection to

```
user_name@ise-pheno-07.auth.ad.bgu.ac.il
```

- Follow the **official setup guides/YouTube** of the tool you choose (VS Code / PyCharm documentation) for SSH remote connection.

Step 5: Work on the GPU Node

Once connected to the GPU node:

You can create files, move files, run files (if you move a lot of data, please use **scp** command instead of using the VS/Pycharm folder GUI because it can crash)

Run your code normally:

Verify GPU usage (you can see the type of the GPU you got):

```
nvidia-smi
```

Run you code normally using the terminal:

```
python train.py
```

How to Stop (Kill) an Interactive GPU Job

If you are done working or your session got stuck, you must manually cancel the interactive job.

Step 1: Find Your Running Job

From the login (master) node (`slurm.bgu.ac.il`), or from your interactive session run:

```
squeue -u $USER
```

You will see your running jobs, for example:

```
JOBID   PARTITION   NAME   USER   ST   TIME   NODELIST
```

```
123456  gtx1080     bash   user   R    02:17  ise-pheno-07
```

Take note of the JOBID (`123456`).

Step 2: Cancel (Kill) the Job

Use the job ID to stop the interactive session:

```
scancel 123456
```

This:

- **Immediately stops the job**
- **Releases the GPU**
- **Frees the node for other users**

Important Note

⚠️ **Always cancel your interactive job when finished.**
if you are not using the GPU you will receive emails that your session will be killed automatically.
Also, leaving jobs running blocks GPUs for other students.

Important Notes

- Interactive GPU usage is **optional** for the course
- Each user is typically limited to **one GPU at a time**
- Always exit your session when finished (or you will get mails that your session will be killed automatically without usage)